Swarm Intelligence in Wavelet Based Video Coding

M. Thamarai and R. Shanmugalakshmi

Karpagam college of Engineering, Government college of Technology, Coimbatore,
India

1. Introduction

Video compression plays an important role in modern multimedia applications such as video streaming, video telephony, video conferencing, etc., A lot of compression algorithms those have been developed are not sufficient for the multimedia applications. In general video coding techniques are classified into two. Discrete Cosine transform (Block based) technique, which is used in the standard compression algorithms such as H.261, H.262, H.264 and wavelet transform based technique. Motion compensated wavelet transform based video coding algorithms are going to be taken as one of the standard compression techniques, since multi resolution capability of the wavelets improves the quality of the signal than that of the DCT based one.

1.1 Necessity of video compression and standards

A low bit rate video coding (bit rate less than 64 kbs) needs high compression ratio (above 150). In high compression ratio video coding, block based coders introduce blocking artifact and ringing effect (Due to Gibbs phenomena) in the reconstructed signal. High compression image coding has triggered strong interests in recent years. In this type of coding, visible distortions of the original image are accepted in order to obtain very high compression factors. High compression image coders can be split into three distinct groups. The first group is called waveform coding and consists of transform and subband coding. The second group called second generation techniques, consisting of techniques attempting to describe an image in terms of visually meaningful primitives (contour and texture, for example). The third group is based on the fractal theory.

An uncompressed video sequence for very low bit rate applications typically requires a bit stream of up to 10 Mbit/s. In order to achieve very low data rates, compression ratios of about 1000 : 1 are required to meet the needs of the large public. Intensive research has been performed in the last decade to attain this objective [Touradjebrahimi et. al, 1995]. Variations of the recommendation H.261 for very low bit rate applications have been defined as simulation models. For these simulation models, severe blocking artifacts occur at very low data rates.

Wavelet based video coding is developing a new area in video coding for last two decades. Because of the multi resolution property, wavelet tool is suitable for image enhancement and compression. Rather than a complete transformation into the frequency domain, as in DCT or FFT (Fast Fourier Transform), the wavelet transform produces coefficient values

which represent both time and frequency information. The hybrid spatial-frequency representation of the wavelet coefficients allows for analysis based on both spatial position and spatial frequency content. While most wavelet-based compression techniques employ the traditional critically sampled discrete wavelet transform (DWT), alternative wavelet transforms have recently been proposed. Specifically, the complex dual-tree discrete wavelet transform (DTCWT) has undergone investigation in 3D video-coding systems [B. Wang, et. al, 2004].

1.2 Particle swarm optimization

Particle Swarm Optimization (PSO) is global optimization technique based on swarm intelligence. It simulates the behavior of bird flocking [Kennedy et. al, 1995]. It is widely accepted and focused by researchers due to its profound intelligence and simple algorithm structure. Currently PSO has been implemented in a wide range of research areas such as functional optimization, pattern recognition, neural network training and fuzzy system control etc., and is successful. In PSO, each potential solution is considered as one particle. The system is initialized with a population of random solutions (particles)and searches for optima (global best particle), according to some fitness function, by updating particles over generations; that is, particles "fly" through the N-dimensional problem search space to find the best solution by following the current better-performing particle. When compared to Genetic Algorithm, PSO has very few parameters to adjust and easy to implement. The variants of PSO's such as Binary PSO, Hybrid PSO, Adaptive PSO and Dissipative PSO are used in various image processing applications.

Recently PSO has been extended to deal with multiple objective optimization problems [K. U. Parsopoulos et. al, 2002]. In the past few years many research works have been focused on modifying PSO to handle multiple objective optimization problems known as multi objective particle swarm optimizer MOPSO. The fixed population size MOPSO and variable population size PSO (Dynamic PSO) are used throughout the evolution process to explore the search space to discover the non dominated individuals (particles). Most of the real life problems are multi objective nature.

Multi objective optimization using PSO has been used in Digital image processing like image segmentation, data clustering etc. Here Video compression also viewed as a muliobjective one. The constraints are Means Square Error (MSE), Computation Time, and Computation complexity and compression ratio. In this chapter the only three constrains (Means Square Error (MSE), Computation Time and compression ratio) are considered for the PSO based optimization. All the three are minimization functions. The fixed population size MOPSO is used throughout the evolution process to explore the search space to discover the non dominated individuals (particles).

First the image is decomposed into subband using the Dual tree wavelet transform and the subband coefficients are minimized using noise shaping method. After that the MOPSO algorithm is used to select the optimum subband which provides less mean square error and desired bit rate. In this MOPSO weighted average approach is used. The constraints total weight age is one. The obtained results are compared with the standard algorithms.

2. Shortcomings in the conventional discrete wavelet transform

Theoretically-sampled form of the wavelet transform (Discrete Wavelet transform DWT) provides the most compact representation; DWT has the following advantages:

- Multi-scale signal processing technique (Both frequency and time resolution).
- DWT transform itself introduces compression.
- Straightforward computation technique.

However, it has several limitations as explained below. It lacks the shift-invariance property and in Image processing applications it has poor resolution in distinguishing the orientations of the object and in multiple dimensions it does a poor job of distinguishing orientations, which is important in image processing. The four drawbacks of DWT are as follows [Ivan. W. Seleshick et. al, 2005].

- wavelet coefficients are oscillatory in nature.
- Since wavelets are band pass filters the wavelet coefficients tend to oscillate between positive and negative singularities as shown in fig. 1 This considerably complicates the singularity extraction and modeling and feature extraction. Shift variant
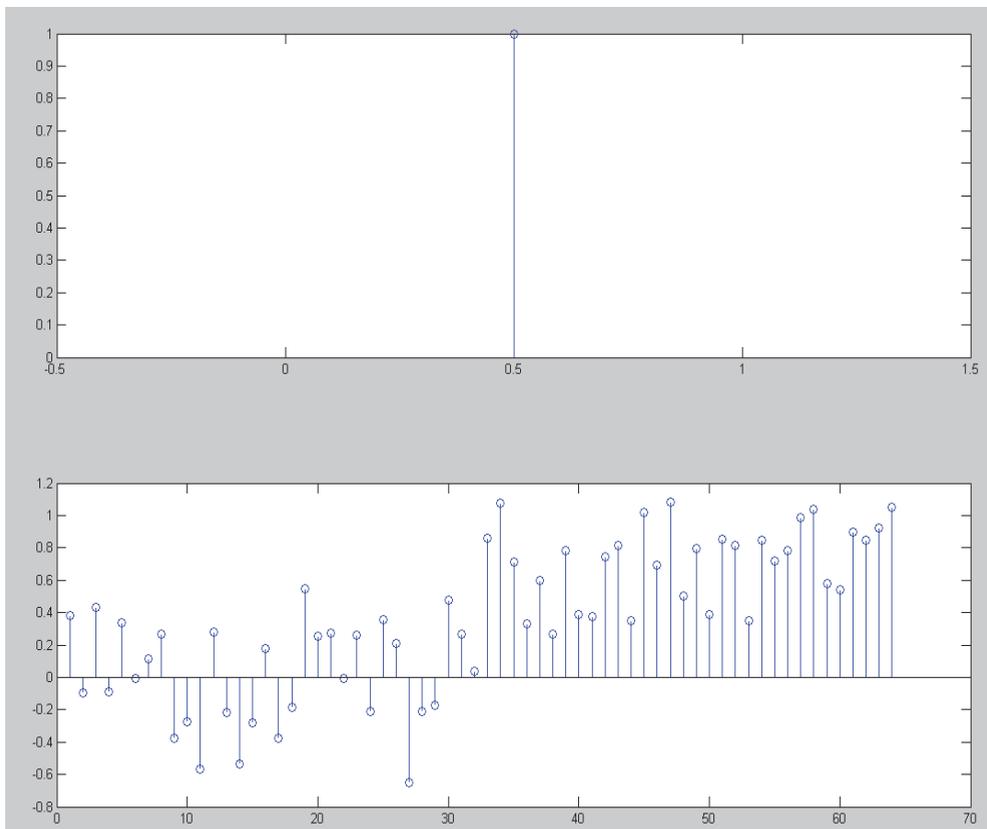

Fig. 1. Oscillatory nature of DWT Coefficients for the signal x(n-0.5)

A small shift of the signal greatly perturbs the wavelet coefficient oscillation pattern around singularities (see Figure 2). Shift variance also complicates wavelet-domain processing; algorithms must be made capable of coping with the wide range of possible wavelet coefficient patterns caused by shifted singularities.
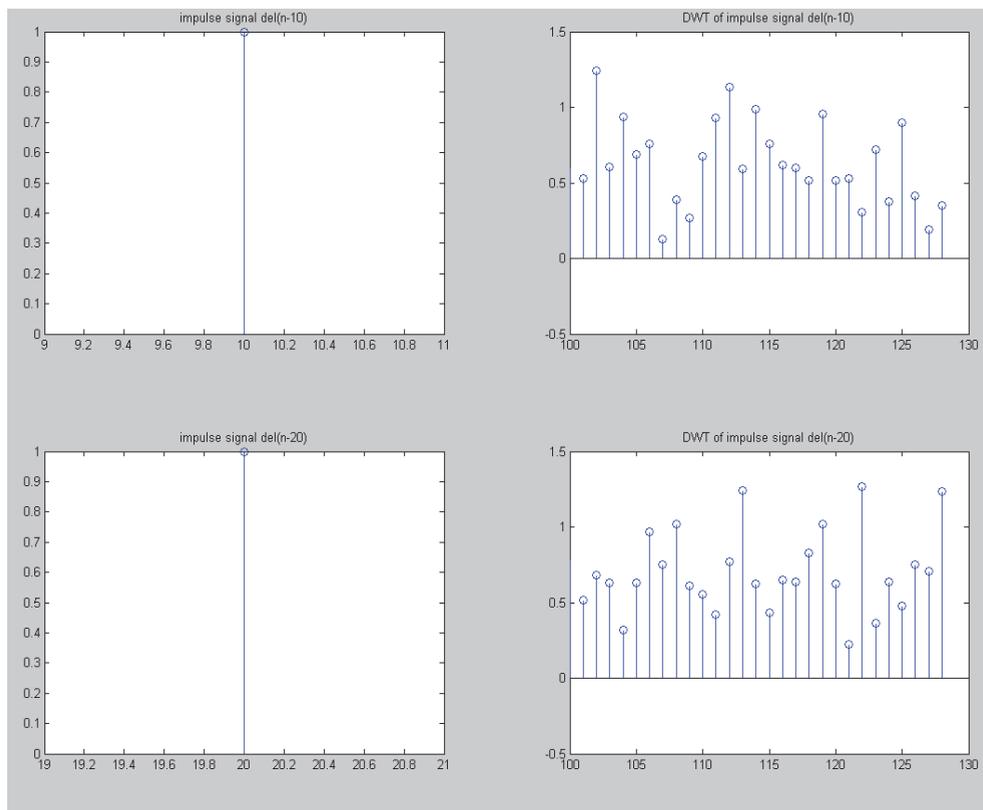
Fig. 2. Shift variant property of DWT

This can be explained with the help of the unit step signal x(n). The x(n) and its shifted version x(n-3) are subjected to DWT and DT CWT decomposition at a level of 3. The spectrum of DWT δ(n-10) is different from the δ(n-20). These two signals are shown in figure 2. Therefore small shift in the input signal results the variation of the DWT coefficients. DWT coefficient values varies based on shifting than DT CWT.

• Aliasing:

The wavelet coefficients of a signal are computed iteratively with the help of sub band decomposition using low pass and high pass filters. These filters are non ideal and results in substantial aliasing. The inverse DWT cancels this aliasing, only if the wavelet and scaling coefficients are not changed. Any wavelet coefficient processing (thresholding, filtering, and quantization) upsets the delicate balance between the forward and inverse transforms, leading to artifacts in the reconstructed signal.

• Lack of directionality:

The multidimensional wavelets produce a check board pattern, and is oriented in several directions. This lack of directional selectivity greatly complicates modeling and processing of geometric image features like ridges and edges.

2D wavelet transform is formed by three wavelets.

3. Dual tree discrete wavelet transform (DTDWT)

It is an expansive type of transform. An expansive type transform is one which converts M number of samples into N coefficients (N>M). According to its name it uses two critically sampled DWTs in parallel for signal decomposition and reconstruction.

3.1 Properties
1. The coefficients of dualtree wavelet transform (1-D) are positive. The dualtree wavelet coefficients of signal x(n-0.5) is shown in figure 3 and are non oscillatory nature.
2. Shift invariant property [Ivan. W. Selesnick et. al, 2005]. The DTDWT of $x(n) = \delta(n − 60)$ and $x(n) = \delta(n − 70)$ are shown in figure 4. The wavelet coefficients of both cases are more or less equal and thus the transform satisfies the shift invariant property.

Directional property: The basis function of dual tree discrete wavelet transform is oriented at a certain direction as +75, -75, +45, -45, +15 and -15. Because of this, check board problems are not present in Dual tree wavelet transforms and no need to estimate motion vectors in a video sequence. Since the transform has multi directional kernels, motion estimation and compensation is a tedious process in the conventional video coding standards. But in the case of DWT, HH band mixes the directions of +45 and -45 together, resulting in check board pattern. The kernels of DWT and Dual tree DWT are shown in figure 5, figure 6 and figure 7.
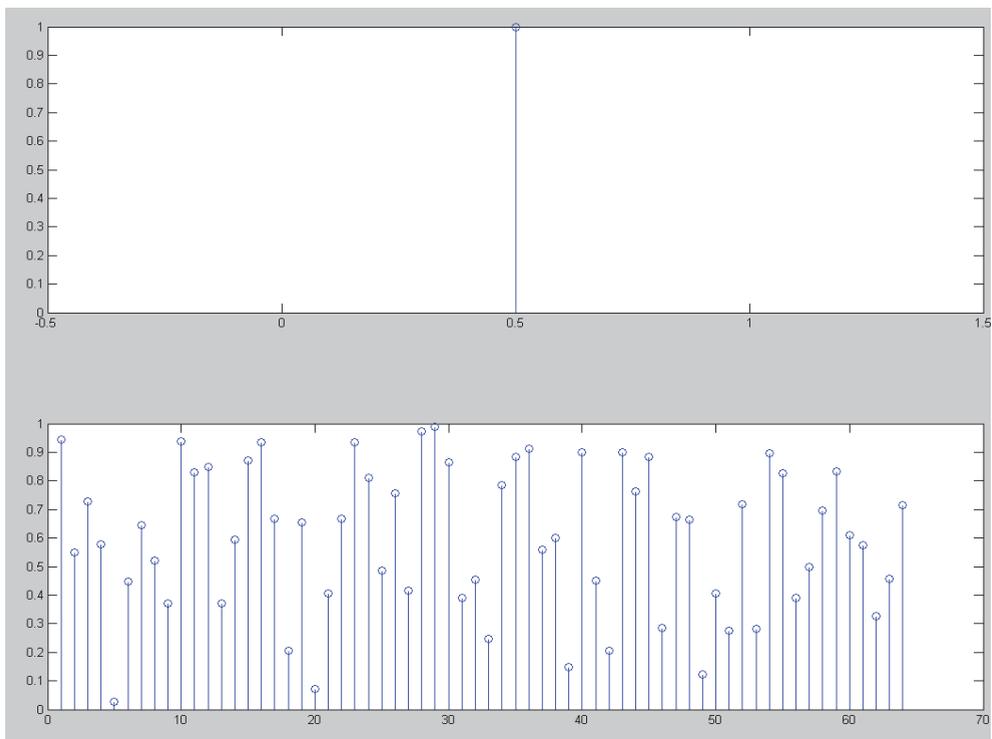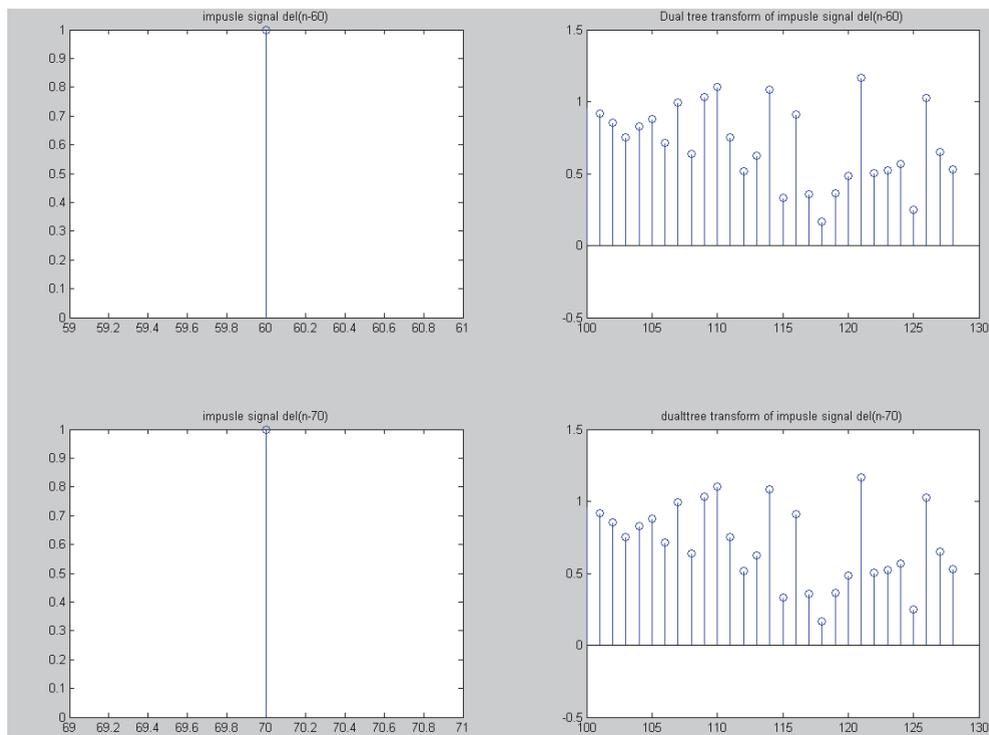

Fig. 3. Signal x(n-0.5) and its dtdwt spectrum

Fig. 4. The two impulse signals x(n) = δ(n − 60) and *x(n) = δ(n − 70)* and their dual-tree complex discrete wavelet transform coefficients
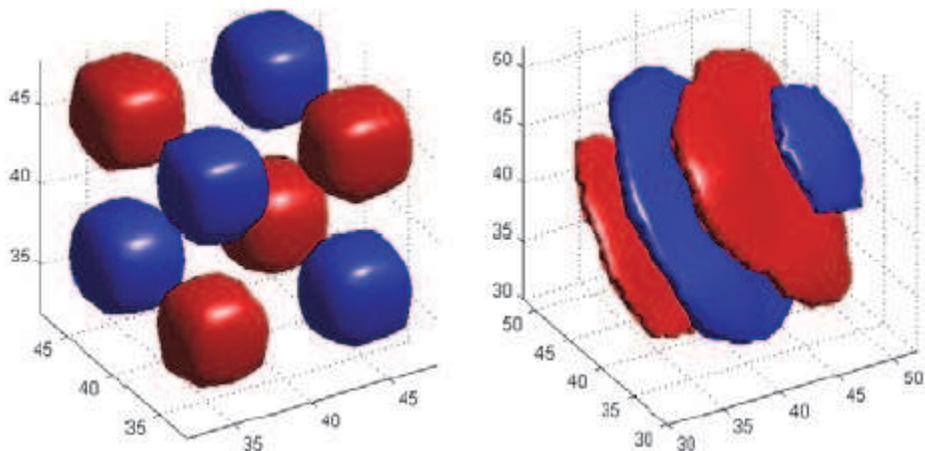

Fig. 5. Kernels of DWT and dual tree DWT

2D dual tree wavelet transform uses six wavelets. The spatial and frequency domain representation of these six wavelets are shown in fig. 7. In both domains the check board

problem is not there. Figure 8 represents the high compact representation property of 2D DTDWT [Ivan. w. Selesnick 2003].
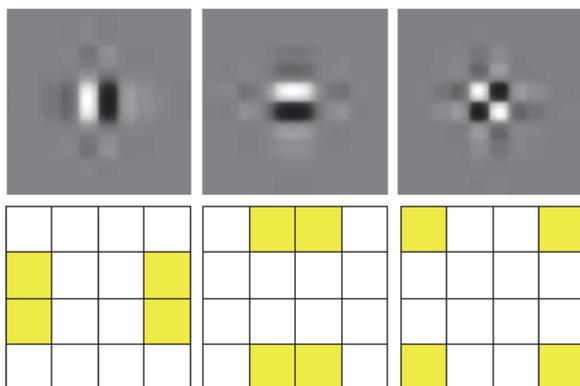

Fig. 6. Three different wavelets. Top row - discrete wavelet transforms in spatial domain. Bottom row - corresponding dwt in frequency domain
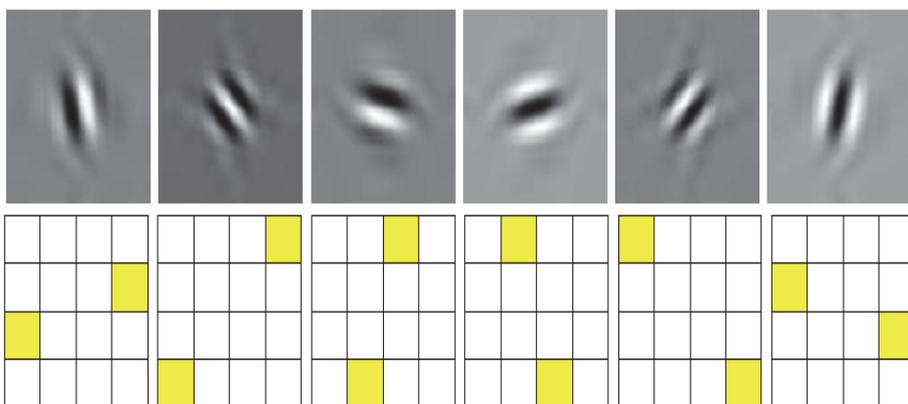

Fig. 7. Top row shows the 2D Dual tree wavelet transform in spatial domain and bottom row shows the same in frequency domain.
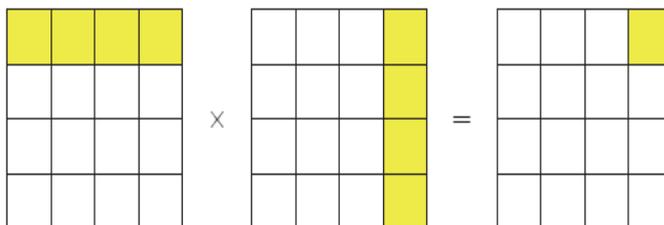

Fig. 8. Compact representation of DTDWT

3.2 Filter bank of dualtree discrete wavelet transform

According to its name the 1 D Dual tree wavelet transform uses two wavelet trees. The input signal is applied to the two trees and it is decomposed into four subband. $h_0(n)$, $h_1(n)$ and $g_0(n)$, $g_1(n)$ are the low pass and high pass filters of the two wavelet trees respectively. The three level decomposition of the transform results in 8 subbands as shown in figure 9. Similarly the synthesis filter bank of 1-D dual tree wavelet transform contains two synthesis wavelet filter banks as shown in figure 10.
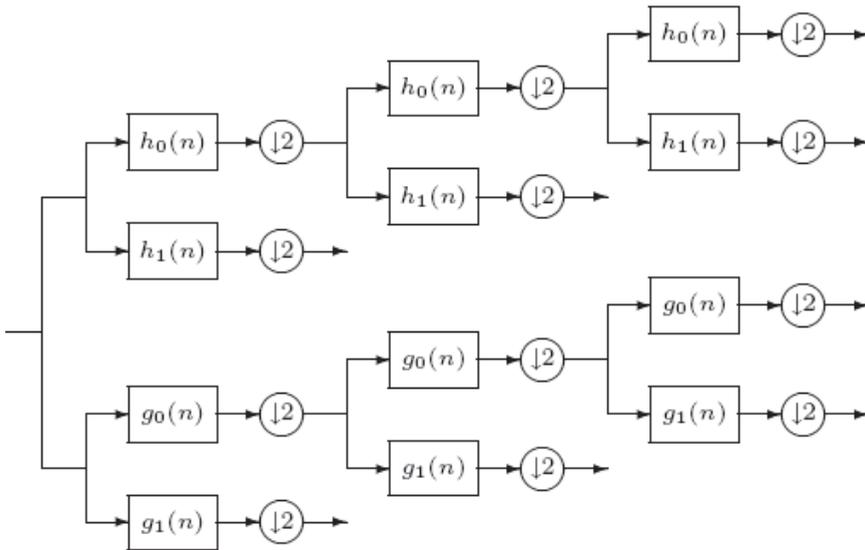
Fig. 9. Analysis filter bank of 1-D dualtree wavelet transform
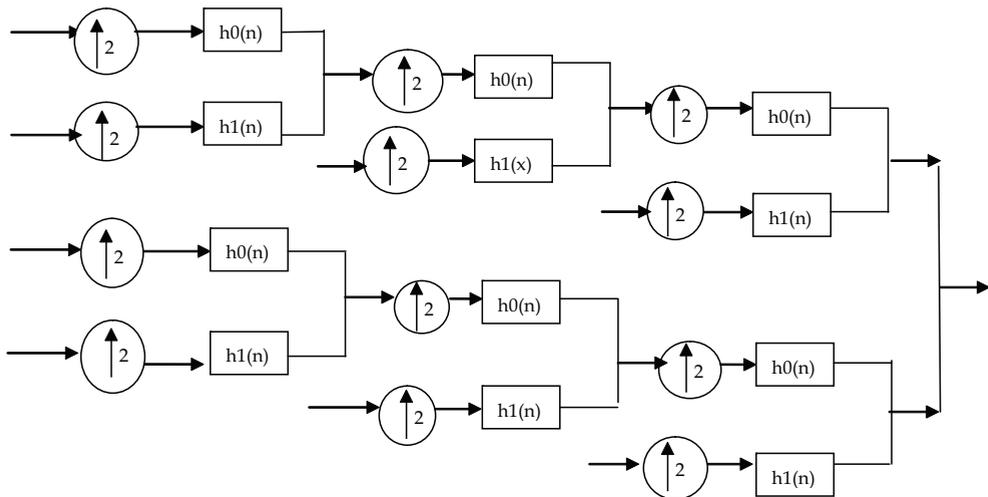
Fig. 10. Synthesis filter bank of 1-D dual tree wavelet transform

3.3 3-D Dual tree wavelet transform

The 3-D wavelet transform is obtained as a combination of four wavelets.

The one dimensional complex wavelet transform is given by

$$\Psi(x) = \Psi h\,(x) + j\Psi g(x) \tag{1}$$

2D dual tree complex wavelet function

$$
\begin{aligned}
\Psi(x,y) &= (\Psi h\,(x) + j\Psi g(x)) + (\Psi h(y) + j\Psi g(y)) \\
&= (\Psi h\,(x)\Psi h(y)) - (\Psi g(x)\Psi g(y)) + j(\Psi g(x)\Psi h(y)) + (\Psi h(x)\Psi g(y))
\end{aligned}
\tag{2}
$$

Real part of

$$\Psi(x,y) = (\Psi h\,(x)\Psi h(y)) - (\Psi g(x)\Psi g(y)) \tag{3}$$

The 3-D separable dual tree wavelet transform is obtained as follows [Ivan. w. Seleslnick et. al, 2003].

The three dimensional complex wavelet is defined as

$$\Psi(x,y,z) = (\Psi h\,(x) + j\Psi g(x))\,(\Psi h(y) + j\Psi g(y))\,(\Psi h(z) + j\Psi g(z)) \tag{4}$$

Real part

$$[\Psi(x,y,z)] = \Psi_1(x,y,z) - \Psi_2(x,y,z) - \Psi_3(x,y,z) - \Psi_4(x,y,z) \tag{5}$$

So it is necessary to take four separable transforms instead of three in the case of 2-D transform.

3-D separable wavelet transform is obtained by the orthonormal combination matrix of four three dimensional wavelet transforms as in equation (6)

$$
\left.
\begin{aligned}
\Psi a(x,y,z) &= 0.5\,[\Psi_1(x,y,z) - \Psi_2(x,y,z) - \Psi_3(x,y,z) - \Psi_4(x,y,z)] \\
\Psi b(x,y,z) &= 0.5\,[\Psi_1(x,y,z) - \Psi_2(x,y,z) + \Psi_3(x,y,z) + \Psi_4(x,y,z)] \\
\Psi c(x,y,z) &= 0.5\,[\Psi_1(x,y,z) - \Psi_2(x,y,z) + \Psi_3(x,y,z) - \Psi_4(x,y,z)] \\
\Psi d(x,y,z) &= 0.5\,[\Psi_1(x,y,z) + \Psi_2(x,y,z) + \Psi_3(x,y,z) - \Psi_4(x,y,z)]
\end{aligned}
\right\}
\tag{6}
$$

where $\psi_1, \psi_2, \psi_3, \psi_4$ are real 3-D wavelets defined in [Ivan. w. Selesnick, et. al, 2003].

By applying this combination matrix to each sub band, the 3-D oriented dual-tree wavelet transform is obtained. The low subbands $\psi_1, \psi_2, \psi_3, \psi_4$ are always positive (because they are low-pass filtered values of the original image pixels), ψ_a is always negative and other three low subbands are all positive. This property of low sub bands can be used to code the sign information of low sub bands very efficiently.

However, [Wang's et. al, 2004] investigation has shown that after noise shaping, 3-D DTDWT needs fewer coefficients than 3-D DWT to achieve the same video quality for all sequences. This result is encouraging and has prompted to explore the use of this transform for video coding.

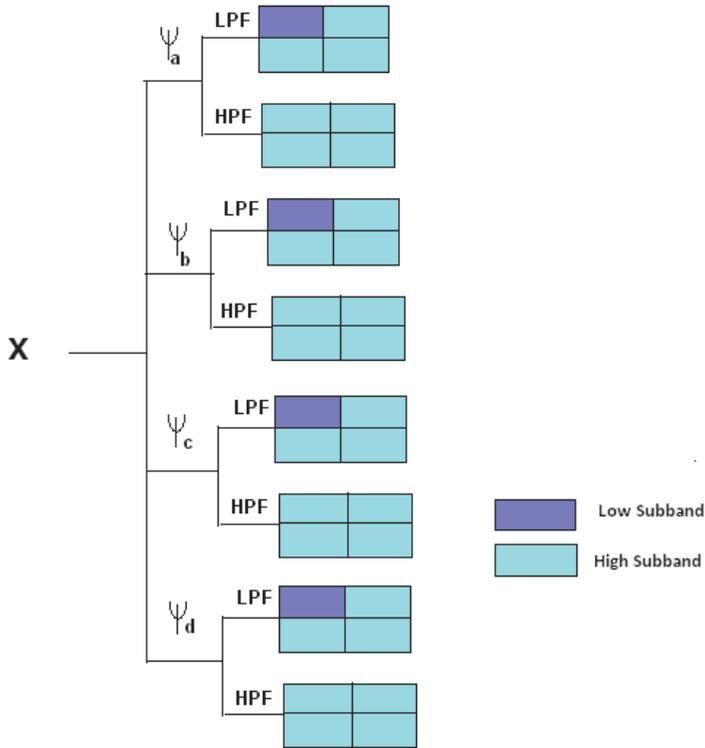The one level decomposition of such a filter is shown in figure 11.

Fig. 11. The 3D Dual tree wavelet transform filter bank with one level decomposition of input signal x

Being an expansive type transform, the number of significant coefficients are identified by noise shaping algorithm [T. H. Reeves et. al, 2002]. In this algorithm, the coefficients are obtained by running the projection algorithm with a preset initial threshold and gradually reducing it until the number of coefficients reach N,a target number. In each iteration the error coefficients are multiplied by a positive real number K and added back to the previously chosen large coefficients to compensate for the loss of small coefficients due to thresholding. This algorithm is applied for video signals and its performance is verified   [Beibei Wang et.al,2004].

The DTDWT is an over complete transform with limited reduncy (2^m:1 for m dimensional signals) Thus for 3D DTCWT the redundancy becomes 8:1 However the real DT DWT reduces the redundancy as 4:1 but with the same motion selectivity. [Beibei Wang et al 2004] proved that without motion compensation (conventional method) the DTDWT performs video coding in a better way.

4. Particle swarm optimization and Multi objective PSO

In PSO algorithm, each individual is called a particle and is subjected to movement in a multidimensional search space to find the best solution. Particles have their own memory, so they retain the part of their previous states. Each particle's movement from initial value

to next position during iteration period is based on their initial velocity and two randomly weighted influences such as Individuality and Sociality

Individuality is the ability to retain the particle's best position and Sociality is the ability to move towards the neighborhood's best previous position. The velocity and position of the particles are updated as follows

$$V_{id}(t+1) = w \times v_{id}(t) + c_1 \times rand_1(.) \times (p_{id} - x_{id}) + c_2 \times rand_2(.) \times (p_{gd} - x_{id}) \qquad (7)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1), \quad 1 \leq i \leq N, \quad 1 \leq d \leq D \qquad (8)$$

where, N is the number of particles and D is the dimensionality; $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$, $v_{id} \in [-v_{max}, v_{max}]$ is the velocity vector of particle i, which decides the particle's displacement in each iteration. Similarly, $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$, $x_{id} \in [-x_{max}, x_{max}]$ is the position vector of particle i which is a potential solution in the solution space. The quality of the solution is measured by a fitness function; w is the inertia weight which decreases linearly during a run; c_1, c_2 are both positive constants, called the acceleration factors which are generally set to 2.0; $rand_1(.)$ and $rand_2(.)$ are two independent random number distributed uniformly over the range [0, 1]; and p_g, p_i are the best solutions discovered so far by the group and itself respectively.

In the t + 1 time iteration, particle i uses p_g and p_i as the heuristic information to update its own velocity and position. The first term in the above equation represents diversification, while the second and third intensification. The second and third terms should be understood as the trustworthiness towards itself and the entire social system respectively. Therefore, a balance between the diversification and intensification is achieved based on which the optimization progress is possible

4.1 Simple PSO algorithm
1. Initialize particles of population size N
2. Find the fitness value of each particle using the defined fitness function
3. Update the velocity and position of each particle according to equations
4. Check the stopping criteria. If it is reached then stop. Otherwise go to step 2

4.1.1 Stopping criteria
There are two types of stopping criteria used.
1. Maximum number of iterations: In certain problems after certain number of maximum iteration there is much more change in the particles position and velocity. After reaching this condition the algorithm stops.
2. Minimum inertia weight: The inertia weight w is reduced from iteration to iteration as follows

$$\left(wmax - \frac{wmax - wmin}{Itermax} \right) Iter \qquad (9)$$

Where Itermax-maximum iteration number and Iter-current iteration

For minimization problems, we specify a very small threshold ε, and if the change of p_g during t times of 4 iteration is smaller than the threshold, we consider the group's best value very near to the global optimum, thus the matching procedure stops.

4.2 Multi objective PSO

Most of the problems in the real world are multi objective in nature. And they have multiple optimum solutions for different objectives. Initially most proposed of the PSO methods deal with single solution. Some problems may have more than one global optimum or both global and local optima need to be located. Therefore some variants have been developed to deal with particular problem with multiple solutions. Niching algorithms have been proposed to deal with particular problem with multiple solutions. Multi objective optimization with particle swarms, called MOPSO is developed to solve the problems those require simultaneous optimization of number of objectives. Many PSO algorithms have been developed under dynamic environments rather than static environment.

The general multi objective optimization problems can be defined in the following format. Optimize

$$f(\overline{x}) = \left[f1(\overline{x}), f2(\overline{x}) \ldots fn(\overline{x}) \right] \text{ where } \overline{x} = (x1, x2, \ldots, Xn)\varepsilon \, \overline{R}n \qquad (10)$$

which satisfies the m inequality constrains $gj(\overline{x}) \le 0$ for j=1,2 …. m and p equality constraints $hj(\overline{x}) = 0$ for j=1,2, …. P the constraints $gi(x)$ and $hj(x)$ define the feasible region Ω and any point in the Ω defines a feasible solution.

For multi objective optimization problems, objective functions may be optimized separately from each other and of the best solution can be found for each objective dimension. However suitable solutions for all the functions can seldom be found. This is because most of the objective functions are in conflict with each other.

The family of solutions of a multi objective optimization problem is composed of all those potential solutions such that the components of the corresponding objective vectors cannot be improved simultaneously. These types of solutions are known as pereto optimal solutions. The population size of MOPSO is larger than the traditional PSO, in order to cover more pereto optimal solutions.

In order to handle multiple objectives, PSO must be modified before being applied to MO problems. In most approaches [Xiaohui et. al, 2003] the major modifications of the PSO algorithms are the selection process of gbest and pbest. In [Cello cello et al, 2004] the paper developed a grid based gbest selection process and also employed a second population to store the non dominated solutions. From the second population, using roulette wheel selection, they selected the gbest randomly. The pbest is selected according to the pareto dominace.

[Parsopoulos et al 2002] used different population for different objectives. It is called vector evaluated particle swarm, n no.of swarms are used to solve n objectives. In their algorithm, when one swarm updates the velocities of the particle the other swarm is used to find the best particle to follow. In another method [Fields et. al, 2002] a new data structure is proposed to cope with the shortcomings of using constant size population. Ans also [Xiaohui et. al, 2003] proposed a method called dynamic neighborhood PSO. In [Xiaohui et al 2003], multiobjectives are divided into two groups F1 and F2. F1 is the neighborhood objective and F2 is the optimization objective. Based on the distance measured, the nearest m particles are grouped as the neighbors of F1 and remaining are assumed as the neighbors of F2. From the grouped neighborhood around F1 the nbest(gbest) is selected. Pbest is the position in a particles history. Whenever the current solution dominates the pbest,only then the pbest is updated.

Evolutionary optimization techniques play an important role in image processing such as Image compression, clustering and object tracking etc., but there has not been much in video coding. In this work we explore the possibility of applying multi objective optimization algorithm to improve the performance of compression algorithm in order to support multimedia applications especially for video. To formulate the mathematical equation to the problem, we consider functions related to the compression rate, computation time and number of frames.

The objective functions describing the MOPSO system for video coding can be represented in figure 12.
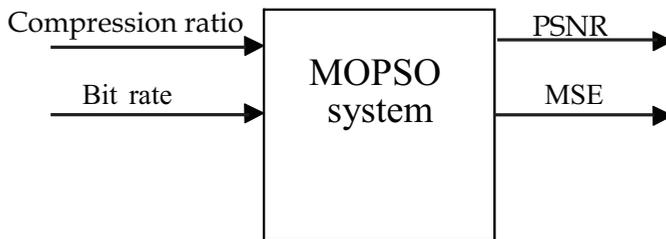

Fig. 12. MOPSO system for video coding

In this MOPSO two swarms are assigned for two objective functions. The objective functions taken are shown in figure 12. The compression ratio and bit rate are considered here as constraints. According to the given constraints the PSNR and MSE are measured.

5. Video coding with Multi objective PSO

The proposed video coder block diagram is shown in figure 13. The input video sequence at the frame rate of 10 frames per second is decomposed (3- levels) using dual tree discrete wavelet transform. There are two sets of filters are used. At level 1, 13-19 near orthogonal filters and at levels ≥ 2, 14 tap Qshift filters are used. The number of significant coefficients is obtained using noise shaping algorithm. Here the initial preset threshold value is set as 220 and is reduced until the number of coefficients reach 40. The energy compensation parameter K is set as 1.8 for better performance. The selected 40 coefficients are considered as particles. These particles are subjected to the MOPSO algorithm. Here fixed population size of 40 is used. The non dominated solutions (individual particles) are selected according to constrains of compression ratio and bit rate.

The conventional linear aggregating function is used to select the global best solution. Here the 40 swarms are simultaneously searched for their individual objective function. The aggregate multi objective function is calculated as follows.

$F(x) = \Sigma\ w_i k_i\ w_i$ –non negative weights Here $w_1 = 0.6$ and $w_2 = 0.4$ The summation of wi is equals to one.

$F1(x)$ = Means square error

$F2(x)$ = computation complexity

Two video sequences foreman and rhinos are tested and their PSNR values of different compression ratios are given in the table 1.
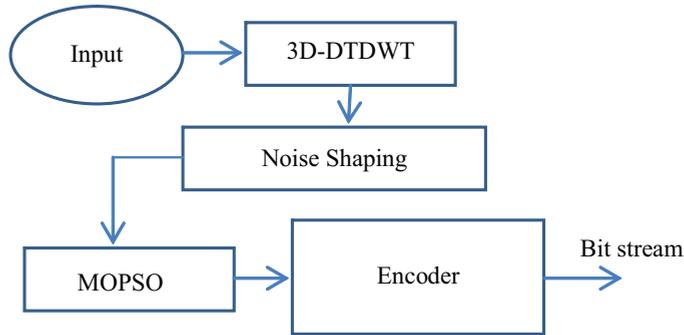
Fig. 13. The block diagram of the proposed system

5.1 Modification in the coder blocks (MOPSO)

In our improved PSO, the new positions are calculated by performing single-point crossover operation with the existing position as performed in GA. This operation avoids the local minima and leads to find the optimum blocks with minimum computational complexity.

where the size of a frame is N × N. The motion estimate quality between the original I_{ogn} and the constructed video sequences I_{cont} is measured in PSNR which is defined as:

$$PSNR = 10 \log_{10} \frac{I_{max}^2}{\sigma_e^2} \tag{9}$$

$$\sigma_e^2 = MSE = \frac{1}{N} \sum_{k=0}^{K} \sum_{i=0}^{N} \sum_{j=0}^{N} \left(I_{ogn}(i, j, k) - I_{cmp}(i, j, k) \right)^2 \tag{10}$$

where K is the number of frames in the video sequence and I_{max} is the maximum gray level value in the frame.

In standard PSO, at each iteration a best position is chosen from the particles known as 'Xpbest'. This best solution is compared with the best solution so far (Xgbest). If current is the best then the global best is interchanged with Xpbest, otherwise the procedure is continued with previous best. In this case, the particles are independent of each other, they are not sharing the information about their travel. This leads to local minimum and may be taking long time for convergence. In our proposed method, we are choosing 'n' number of best solutions at each iteration. The value should satisfy the condition: 1 < n < N/2, where N = total number of particles.

And these 'n' best solutions are compared with previous best solutions. Finally 'n' best solutions are chosen globally. And with these 'n' best solutions are matted with each other at random to fill the population size. Here, we are performing single-point crossover operation to perform matting. For example, if n = 3, and the population size is 5, consider the following 3 best solutions.

<div align="center">
1 0 1 0 0 1 0 1 0 1 0 1 0
1 0 0 0 0 1 1 1 0 0 0 1 1
1 0 1 0 0 1 0 1 1 1 0 0 1
</div>

With these 3 best solutions, to fill the population we need 2 more series that can be generated by performing single point crossover between 2 & 3 as given below.

(2nd best)   1 0 0 0 0 1 1 <u>1 0 0 0 1 1</u>

(3rd best)   1 0 1 0 0 1 0 <u>1 1 1 0 0 1</u>

(Crossover at 5th bit)

(4th solution)   1 0 1 0 0 1 0 1 0 0 0 1 1

(5th solution)   1 0 0 0 0 1 1 1 1 1 0 0 1

And we have introduced a parameter called Velocity Rate (VR) to control the updation of velocity based on the performance history of the particles. Initially all particles are assigned 1 as velocity rate. At each iteration, based on the fitness value the VR is either increased or decreased by 0.1 for each particle. If VR value of the particle which gives the optimum value will be increased to 0.5, and the updated velocity is multiplied with VR. The equation (1) is modified as

$$V_{id}(t+1) = w \times v_{id}(t) + c_1 \times rand_1(.) \times (p_{id} - x_{id}) + c_2 \times rand_2(.) \times (p_{gd} - x_{id}) + VR \qquad (11)$$

And we are introducing one more parameter in our modified PSO, which is the direction (angle) of the particles. Here we have eight different directions, 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315°, from these the particles can choose any one direction at random to select the optimum value, but the condition is that, all the particles have to move in the same direction. With these novel parameters, the PSO can avoid premature convergence.

Algorithm:

Step 1. Generate initial population of Swarm (Xi) and Velocity (Vi).

Step 2. Each swarm represents subset of blocks.

Step 3. Calculate the fitness value of each swarm.

Step 4. Calculate 'n' number of Xgbest for each particle.

Step 5. Change the position and velocity of each particle based on crossover operator and modify velocity rate.

Step 6. Again calculate the fitness value of each swarm.

Step 7. Find out 'n' number of Xpbest for each particle.

Step 8. Evaluate the objective function(Weighted average g approach)

Step 9. Compare Xgbest and XPbest, hold best as Xgbest.

Step 10. Repeat steps from 3 to 9 until the stopping criteria

The algorithm for the general MOPSO system:

Step 1. MOPSO()

Step 2. Iinitialize swarms()

Step 3. Iteration=1

Step 4. White iteration<maximum iteration do

Step 5. Fitness function evaluation

Step 6. Update position and velocity

Step 7. Calculate objective vector

Step 8. Update non dominant set()

Step 9. Iter=iter+1

Step 10. End while

Video Sequences	Foreman			Rhinos		
Compression Ratio	1:4	1:3	1:2	1:4	1:3	1:2
Bit-rate(kbps)	730	1000	1424	730	1000	1424
DTDWT	31.79	32.45	34.51	26.95	29.62	31.15
DTDWT+PSO	33.90	34.90	35.90	28.66	33.11	34.97
DTDWT+MOPSO	34.01	37.43	40.45	29.11	33.94	35.98
3D-SPIHT	29.32	31.47	33.86	26.42	27.29	30.93

Table 1. Average PSNR comparison at different bit rates and compression ratios.
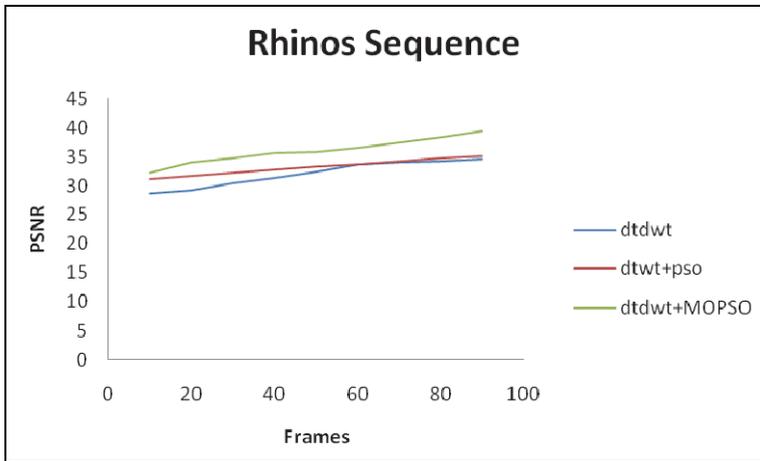
Fig. 14. Performance analysis of Rhinos Sequence.
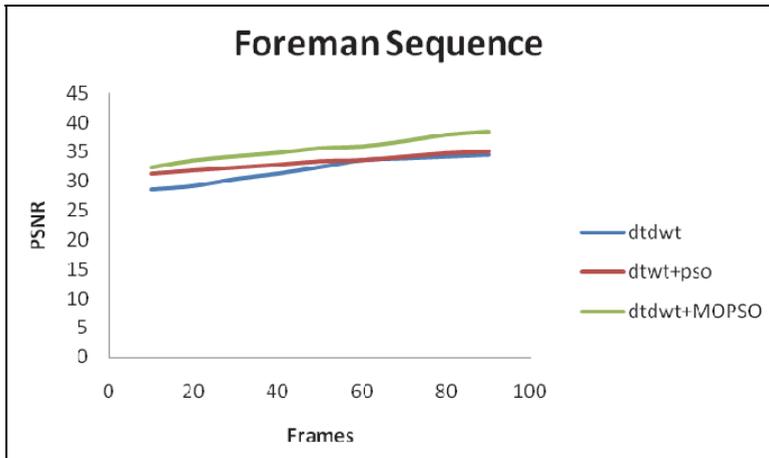
Fig. 15. Performance analysis of Foreman Sequence.

Two video sequences "Foreman" and "Rhinos" are used for testing. Both sequences have 80 frames with a frame rate of 30 fps. Table 1 lists the average PSNR of the two sequences at different bit rates. For a video sequence which has many edges and motions, such like "Foreman", DTDWT+PSO outperforms 3-D SPIHT more than 4 dB. For sequence "Rhinos", DTDWT+PSO offers around 2 dB better PSNR results. Whereas our proposed codec DTDWT+MOPSO outperforms better than DTDWT, 3-D SPIHT with more than 3 dB for both the sequences. Subjectively, DTDWT+MOPSO has better performance than the existing and has the redundancy caused by symmetric extension, the coding results are very promising. Figures 14 and 15 show the performance of the system(PSNR value) with increasing number of frames. The proposed system provides constant PSNR with increasing number of frames.

6. Conclusion

The excellent directional and shift invariant property of Dual tree discrete wavelet transform is used for video coding without motion estimation and compensation. The optimum subband coefficients are selected using multi objective pso with the objective factors of compression ratio and bit rate. The n best solutions particles are selected by means of modification in the velocity rate and incorporating the directional properties of the subbands of DT DWT and crossover among the best solutions. In the multi objective PSO, the pareto optimal solutions are selected based on the constraints and weighted average approach. The performance of the proposed method is measured in terms of PSNR. The PSNR value of this combination(DTDWT+MOPSO) is better when compared to the other methods. In future, by analyzing the inter and intra correlations among the subbands of DTDWT and also by adding the constraints of minimum computation complexity and computation time in the MOPSO, the system performance can be improved.

7. References

Touradjebrahimi, Emmanuel Reusens and Wei Li (1995) *New trends in very low bit rate video coding*" Proceedings of the IEEE

B. Wang, Y. Wang, I. Selesnick, and A. Vetro, (2004) *An investigation of 3D dual-tree wavelet transform for video coding*, in Proceedings of the International Conference on Image Processing, vol. 2, Singapore, pp. 1317–1320.

Kennedy, J., Eberhart, R. C(1995) *Particle swarm optimization*, in IEEE International Conference on Neural Networks, pp. 1942-1948.

K. U. Parsopoulos, M. N. Varahatis (2002), *Particle swarm optimization method in multi objective problems*, Proceedings of the 2002 ACM Symposium on applied Computing Madrid, Spain, ACM Press pp. 603-607

Ivan. w. Selesnick, Richard Baraniuk, Nick G. Kingsbury (November 2005) *The Dualtree Complex Wavelet transform- A coherent framework for multiscale signal and image processing*, IEEE Signal processing Magazine pp. 123 – 151

Ivan. w. Selesnick, Ke. Yong Li (August 2003) *Video denoising using 2D and 3D Complex dualtree wavelet ransform, Proceedings, Wavelet applications in signal and image processing X* SPIE 5207.

Beibei Wang, Yao Wang, Ivan Selesnick & Anthony Vetro (December 2004) *Video coding without motion compensation using a 3 D Dualtree wavelet transform, Mitsubishi Electric Research Laboratories Inc., 2004*

T. H. Reeves and N. G. Kingsbury (September 2002), *Over complete image coding using iterative projection-based noise shaping,* Signal Processing Group University of Cambridge U.K.ICIP 02, Rochester, New York,

Xiaohui Hu Russell C.Eberhart & Yuhui Shi (2003), *Particle Swarm with extended memory for multi objective optimization,* Prudue University, Indianapp. 193-197. IEEE Proceedings.

C. A. Cello, G. T. Pulido & M. S. Leehuga (2004), *Handling multi objective with particle swarm optimization*, IEEE evolutionary computing 2004

Fieldsend & J. E. Singh (2002) *A multi objective algorithm based upon particle swarm optimization, an efficient data structure and Turbulance,* Proceedings of the 2002 U.K. workshop on computational Intelligence, Birminham U.K. pp. 37-44.
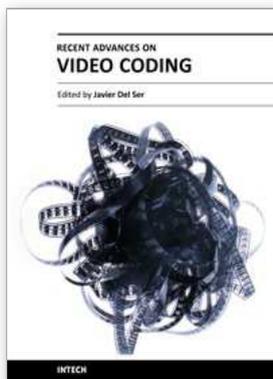
Recent Advances on Video Coding

Edited by Dr. Javier Del Ser Lorente

ISBN 978-953-307-181-7

Hard cover, 398 pages

Publisher InTech

Published online 24, June, 2011

Published in print edition June, 2011

This book is intended to attract the attention of practitioners and researchers from industry and academia interested in challenging paradigms of multimedia video coding, with an emphasis on recent technical developments, cross-disciplinary tools and implementations. Given its instructional purpose, the book also overviews recently published video coding standards such as H.264/AVC and SVC from a simulational standpoint. Novel rate control schemes and cross-disciplinary tools for the optimization of diverse aspects related to video coding are also addressed in detail, along with implementation architectures specially tailored for video processing and encoding. The book concludes by exposing new advances in semantic video coding. In summary: this book serves as a technically sounding start point for early-stage researchers and developers willing to join leading-edge research on video coding, processing and multimedia transmission.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

INTECH
open science | open minds